Timex Sinclair User Group NEWSLETTER

SEPT 1987 Ottawa Chapter

EDITOR - M. Dove Volume IV Issue II

TABLE OF CONTENTS

Meeting/Ad Rates 0001

From the Editor.......... 0010

From the Library......... 0010

Meetings in Review....... 0011

2068 Programs............ 0100

ADVERTISING RATES

Members - 50 words or less.
(No charge)

Non-Members - 50 words or less
($5.00)

Commercial

Full page	*$30.00*
Half page	*$22.00*
Quarter page ..	*$12.00*
Eighth page ...	*$ 8.00*

MEETINGS

LOCATION: All meetings are held at the NRC bldg.. 100 Sussex Dr. Ottawa. Check with security when signing in for the room number.

TIME:Meetings are held at 8.00 P.M. the second Tuesday of every month (no meetings in July or August).

FORMAT:

1)-President's Address and introduction of new members.
2)-Software review (max of 20 minutes).
3)-Hardware review (max of 20 minutes).
4)-Adjourment of the formal part of the evening.
5)-Informal: Members discuss common problems, swap programs, and exchange info (tips) etc.

MEETING DATES

April	14th
May	12th
June	9th
September	15th
October	13th
November	10th
December	8th

MAILING ADDRESS

Please Address all correspondence to:

Michael Dove
The Editor
OHTSUG
2161 East Acres Rd.,
Gloucester, Ontario
Canada K1J 9A6

FROM: THE EDITOR

In case you do not have our numbers then read on.

President - Alan Moore
(226-7118)
Admin. - Michael Dove
(745-1779)
Software - Bill Harmer
(722-7184)
Hardware - John Mathewson
(746-7669)
Librarian - David Solly
(731-2120)
Treasurer - Iris Ten Holder
(828-6668)

Letter form the Librarian

by
David Solly

I would like to take this opportunity to welcome new members and to welcome back old members. I hope everyone had an enjoyable summer.

During the summer the computer stork came by the library and dropped off a bouncing baby TS 1000 disc dirve system. Its arrival heralds some changes in the library.

1) Any user who has a Larken disc drive for either TS 2068 or the TS 1000 will be able to borrow programs in disc format.

2) There are now a number of Larken specific programs and utilities available in the library.

3) Circulating lists of all the prgrams available from the library have been printed. All included there are close to 200 programs listed.

4) Because there are more programs in the library than tape to record them on, the circulating tape collection will be limited to recent titles. All the other programs have been backed up on disc and are still available.

5) If anyone wishes a tape made of a program which is not presently in the circulating tape library, or, wishes to reserve a program, he or she may do so by filling a request form at meetings or by phoning me at 731-2120 evenings between 7:30 and 9:00. Deadline for requests is the weekend before the meeting.

6) Blank tapes will be available from the library for those who wish to record programs for donation. These blank tapes will be charged out as a regular circulating items.

New Arrivals!

Thanks to the efforts of Bill Harmer we have recieved a tape containing some 30 programs for the TS 2068 and the ZX Spectrum (aka TS 2000) from the Timex Sinclair user group in Long Island, New York. Among these are some striking sound and graphics programs. From George Chambers of the Toronto Timex Sinclair user group have come a number of programs and utilities for the Larken disc drive system as well as a number of other programs. A series of electronics programs for the TS 1000 were donated last meeting by J. Bateson. Other programs for the TS 1000 have been donated by Michael Dove, Ken Schieman, Bill Harmer and myself.

On the Book Scene

New books which have arrived are: Using and programming the Timex Sinclair Computer (TS 1000) by Ken Knecht, Color Graphics for the TS 2068 by Nick Hampshire, The Timex Sinclair 2068 by Roger Valentine, and The Sinclair ZX81: programming for real applications by Randle Hurley.

On the Periodical Scene

This June saw the last issue of the British ZX Monthly Magazine. This was the last Sinclair publication available from Britain which still supported the ZX81. During the last months we have received new issues of Your Sinclair, Time Designs, Sinclair User and SyncWare News.

A number of newsletters, some old and some new, are available from the library.

Sincus News (Johnson City, NY), SMUG (Sinclair Milwaukee Users Group), ZX-Appeal (Vancouver BC), SincLink (San Franscisco Bay Area Sinclair Technology User Group), T/S Users Group of Vancouver, and Timex-Sinclair Program Bulletin, Ottawa Support Unit.

I hope to be available at least one evening in the week for people who would like to come to my place to borrow material and consult books and periodicals. Give me a call at the time and number mentioned above to check if I'll be there. There is also an answering machine at that number so people can leave a message if I am not there.

MEETINGS IN REVIEW

For those of us who are not always able to attend our monthly meetings you might be interested in a summary of the highlights of those meetings (April through June 1987).

At our April meeting the were the following highlights. John Mathewson told us of his woes with a faulty power supply which blew his ROM. John eventually bought a chip and burnt in the ROM sequence. It cost him about $68.00 to get his system up and running. Should any one need a new ROM John has safely tucked away a copy of the 2068 ROM. John is still working on the keyboard for the ZX81 (analogous to the system for the 2068). See my comments in the April newsletter.

Mike Dove gave a demo on four way pixel scrolling. The program designed for the Spectrum works equally well on the 2068. The listing of this program is found later in this newsletter.

Dave Solly (our Librarian) is very interested in the construction of easy to read listings for personal use or for publication. He has adapted a variety of programs to produce neat archival records. Whether the program originally comes from a ZX81 or 2068 the listing can be manipulated via MSCRIPT. The results are quite professional. The various programs to do the manilpulation are listed in this newsletter.

Harry Rosewarne gave a very interesting Demo on the techniques and usefulness of Integers, truncation and rounding off. Perhaps we will see something on his demo in future newsletters.

In June John Mathewson displayed his prototype midi interface. This coupled with the approrpiate software transformed his system into quite an elegant Speech Synthesizer.

Iris Ten Holder gave a demo on her Payroll program. Iris uses this program in her own business to tally receipts, paystubs, and income tax records. At some point she plans to generalize this program to make it more useful for small business persons.

Dave Solly gave an interesting Demo om the use of alternate character sets, thier screen display and the print out via alternate fonts.

2068 PROGRAMS

This program was taken, without modification from the December 1986 issue of Sinclair user (Hewson Helpline). It was designed for the Spectrum but works equally well on the 2068. Yhis program will scroll one pixel at a time in anyone of four directions.

```
  10 FOR n=1 TO 132
  20 READ a
  30 POKE 32767+n,a
  40 NEXT n
  50 STOP
 100 DATA 33,0,64,6,191
 110 DATA 197,229,209,36,124
 120 DATA 230,248,188,32,16
 130 DATA 6,8,144,31,31
 140 DATA 31,103,1,32,0
 150 DATA 9,124,23,23,23
 160 DATA 103,229,1,32,0
 170 DATA 237,176,225,193,16
 180 DATA 220,24,43,33,224
 190 DATA 87,6,191,197,229
 200 DATA 209,124,230,248,188
 210 DATA 40,3,37,24,16
 220 DATA 14,7,129,31,31
 230 DATA 31,103,1,224,255
 240 DATA 9,124,23,23,23
 250 DATA 103,229,1,32,0
 260 DATA 237,176,225,193,16
 270 DATA 218,54,0,229,209
 280 DATA 19,14,31,237,176
 290 DATA 201,33,0,64,6
 300 DATA 192,197,167,6,32
 310 DATA 203,30,35,16,251
 320 DATA 193,16,244,201,33
 330 DATA 255,87,6,192,197
 340 DATA 167,6,32,203,22
 350 DATA 43,16,251,193,16
 360 DATA 244,201
1000 FOR n=32 TO 255
1010 PRINT  CHR$ n;
1020 NEXT n
1030 FOR n=1 TO 100
1040 RANDOMIZE  USR 32768
1050 RANDOMIZE  USR 32811
1060 RANDOMIZE  USR 32864
1070 RANDOMIZE  USR 32882
1080 NEXT n
```

LINE LISTING
BASIC2 AND M-SCRIPT
A SUPRISE COMBINATION

Article by
David Solly

If you have written many programs for which you have made a line listing for safekeeping or if you are involved, as I am, in the creation of a Sinclair user group news letter you are probably frustrated with the fact that a line listed program does not lend itself easily to cutting, tearing or any other form of editing without damaging some vital bit of information. If you are like me most of your line listings have been folded and stashed in no particular order in a box and the program instructions, if not included in REM statements within the listing, are kept in a file and perhaps notes on modifications in still another file. You have probably wished that there was some way to have all the program documentation together in one file. For me there seemed to be no way out of this dilemma until I encountered a program called Basic2 by Michael E. Carver which appeared in the November/December 1986 issue of Time Designs Magazine.

The original idea behind Mr. Carver's Basic2 program was to create from a Timex-Sinclair 2068 listing an ASCII text data file which could be uploaded to an electrionic bulletin board and which could also be read by owners of other makes of computers which are unable to decode Sinclair BASIC tokens. Once I found that the text file created by Mr. Carver's Basic2 program could be loaded into M-Script [***NOTE: You must use Jack Dohany's version of M-Script. The earlier version with its unique I/O routine will not work!] I realized that now I had a method whereby I could edit and print the listing, whether for archiving or as part of a news letter, in any format I desired. For instance: an article can be written around the listing, as I have done here, the line length can be altered (with no splitting of words!) to suit a TS 2040 printer or a full sized printer. You can divide the listing neatly into pages without chopping any information, double space and add headers, footers, page numbers and so on. But why talk when I can show you. Below is a listing for Michael E. Carver's Basic2 program created as I have outlined.

Basic2
Program Listing

```
  1 LIST 100
 10 REM SAVE ROUTINE
 20 SAVE "Basic2.SC" LINE 1
 30 SAVE "Basic2.SC" LINE 1
 35 CLS : PRINT "Rewind and play to verify"
 40 VERIFY "Basic2.SC"
 50 VERIFY "Basic2.SC"
 60 LIST 100
 70 STOP
 80 REM
 90 REM
100 REM The source of this program is:  Time
Designs Magazine, v. 3, n. 1, Nov/Dec 1986,
p. 31-32.
110 REM This progam was created by Michael
E. Carver to decode Timex Sinclair 2000/2068
BASIC  Listings into ASCII Text files.
120 REM
130 REM
200 REM *** Instructions
210 REM Create and save the machine code by
typing RUN 9000 and following the
instructions.
220 REM Load the BASIC progam to be
converted into ASCII text.
230 REM "Clear 64899 and load the machine
code program at 64900. Execute the first step
of the program with: PRINT USR 64909
235 REM "*** DON'T PANIC! When you hit the
next key the program will seem to have
crashed and you will see the Timex-Sinclair
Logo at the bottom of the screen.
240 REM "Execute the second part of the
conversion with PRINT USR 65042. Save the
resulting ASCII text file as per
instructions.
250 REM
260 REM
9000 CLEAR 64899: LET a$="": RESTORE : FOR
i=0 TO 73: READ d$: LET a$=a$+d$: NEXT i
9010 IF LEN a$<>1184 THEN PRINT FLASH
1;"Error in Machine code DATA      Lines
9925-9999"''"Please correct before
continuing": STOP
9020 LET address=64900: FOR i=1 TO LEN a$-1
STEP 2
```

```
9030 POKE address+INT ((i-1)/2),(CODE
a$(i)-(48 AND CODE a$(i)<58)-(55 AND CODE
a$(i)>64))*16+CODE a$(i+1)-(48 AND CODE
a$(i+1)<58)-(55 AND CODE a$(i+1)>64)
9040 NEXT i
9050 CLS : PRINT "Machine code has been
loaded    into memory."''"Press any key to
save & verify Basic2.C1": PAUSE 0: SAVE
"Basic2.C1"CODE 64900,592: CLS : PRINT
"Rewind and play tape to verify": VERIFY
"Basic2.C1"CODE 64900,592
9925 DATA "0000000000000000"
9926 DATA "00ED4B535C2A4B5C"
9927 DATA "ED422323E5C1ED43"
9928 DATA "89FDED5BB25CED53"
9929 DATA "87FD2A4B5CEDB813"
9930 DATA "ED5385FD11D8FD01"
9931 DATA "3A00CDDB21CDB002"
9932 DATA "7BFEFF20F87AFEFF"
9933 DATA "20F3CDB0027BFEFF"
9934 DATA "28F811006FED53B2"
9935 DATA "5CCD1D0D16000042"
9936 DATA "4153494320686173"
9937 DATA "206265656E206D6F"
9938 DATA "7665642E20160200"
9939 DATA "505245535320414E"
9940 DATA "59204B455920544F"
9941 DATA "20434C454152204D"
9942 DATA "454D4F52592E2AB2"
9943 DATA "5C228BFD2184FD36"
9944 DATA "002A85FD56235E23"
9945 DATA "2323E5E5C12A87FD"
9946 DATA "AFED42D2AEFEE111"
9947 DATA "76FE013700CDDB21"
9948 DATA "2A8BFD11006FED52"
9949 DATA "2B111027CD69FE11"
9950 DATA "E803CD69FE116400"
9951 DATA "CD69FE110A00CD69"
9952 DATA "FE110100CD69FE3E"
9953 DATA "FDCD3012C9AF3CED"
9954 DATA "52380218F919C62F"
9955 DATA "D7C9160000546F20"
9956 DATA "7361766520424153"
9957 DATA "4943206173207465"
9958 DATA "78742066696C653A"
9959 DATA "0D0D534156452022"
9960 DATA "6E616D652220434F"
9961 DATA "4445203238343136"
9962 DATA "2C00EB1E2001E803"
9963 DATA "CDB6FF016400CDB6"
9964 DATA "FF010A00CDB6FF7D"
```

```
9965 DATA "FE202802C630CDC9"
9966 DATA "FFE17E23E5FE2220"
9967 DATA "0AF53A84FDEE0132"
9968 DATA "84FDF1FE0D2817FE"
9969 DATA "0E2009E1010500ED"
9970 DATA "4AE518DDFE203002"
9971 DATA "18D7FE7B302AF53A"
9972 DATA "84FDCB8F3284FDF1"
9973 DATA "FE0DCC19FFFE3ACC"
9974 DATA "19FFCDC9FFFE0D20"
9975 DATA "B8E1C320FEF53A84"
9976 DATA "FDCB973284FDF1C9"
9977 DATA "FE80301DFE7C280F"
9978 DATA "FE7E280B2184FDCB"
9979 DATA "4620C3CB5620BFD6"
9980 DATA "1FCD77FFCD6EFF18"
9981 DATA "88FE9030043E2018"
9982 DATA "ADFEA53004D64F18"
9983 DATA "A5FEEA200AF53A84"
9984 DATA "FDCBD73284FDF1D6"
9985 DATA "A5CD77FFCD6EFFC3"
9986 DATA "CDFE3A84FDCBCF32"
9987 DATA "84FDC9119800F5CD"
9988 DATA "7C07380C3A84FDCB"
9990 DATA "4F20053E20CDA5FF"
9991 DATA "1AE67FCDA5FF1A13"
9992 DATA "8730F5D1FE482803"
9993 DATA "FE82D87AFE03D83E"
9994 DATA "20D5D9CDC9FFD9D1"
9995 DATA "C97BFEFF200E3E00"
9996 DATA "180AAFED423C30FB"
9997 DATA "093D28EDFE202805"
9998 DATA "C63011FF00E52A8B"
9999 DATA "FD7723228BFDE1C9"
```

There are a few details that should be noted when creating a line listing in this manner. Firstly, be sure that the line justification function in M-Script is off when printing the section containing your listing otherwise you will have lines looking like:

```
100            GO                    SUB             2068
```

which you will want to avoid. Secondly, Mr. Carver's program does not, unfortunatly, convert the Sinclair Block Graphics which are found from CHR$ 129 to 143. If block graphics play a vital role in your program you may want to convert them into "printable" characters my merging and running my Block Graphics Blaster program printed below before converting to ASCII text with Basic2. This program scans each line of the

Sinclair BASIC program and replaces any block graphic it finds with a unique character between CHR$ 50 to 64. It also gives the total number of block graphics converted which you may want to add to your program in a REM statement as a reminder before conversion into ASCII text but usually the context of the program is enough of a guide.

Block Graphic Blaster
Program Listing

```
9992 LET n=26715: LET c=0: CLS : PRINT AT
5,0;"Block Graphics Replacer Program": REM
Created by David Solly, Ottawa, Ontario,
Canada.
9993 FOR i=n TO PEEK 23627+256*PEEK 23628
9994 IF PEEK i=13 THEN LET a=256*PEEK
(i+1)+PEEK (i+2): PRINT AT 10,5;"Working on
Line: ";a: IF a>=9999 THEN PRINT AT
18,14;"DONE!": STOP
9995 IF PEEK i=13 OR PEEK i=14 THEN LET
i=i+5
9996 LET a=PEEK i: IF a>=129 AND a<=143 THEN
POKE i,(a-79): LET c=c+1: PRINT AT
12,0;"Total Number of Block Graphics",AT
13,10;"Replaced: ";c
9997 NEXT i
9999 REM This program will search through a
BASIC listing for Block graphic characters
and replace them with a unique character
between ASCII 50 to 64.
```

Once I was satisfied with what I could do with TS 2068 listings I began to wonder what could be done with TS 1000/ZX81 program listings. I realized that this could be best done by first modifying the upload program before uploading the TS 1000 program to the TS 2068 and converting into ASCII text. I went to my collection of Syntax and dug up the article on David Ornstein's First_Loader program. By using the infromation provided by Mr. Ornstein on how to make adaptations to First_Loader I made the following modifications to create my own version of First_Loader.

1) Inverse "A" to "Z" where changed to upload as lower case "a" to "z" and inverse space (or graphic space) was changed to upload as "@". The other inverse character were left to be uploaded as their normal counterparts. This causes few problems since most of the remaining inverse characters are used within some sort of context

or can be indicated by an annotation once the ASCII text version of the listing has been loaded into M-Script.

2) All the block graphics, with the exception of graphic space noted above, where changed to be represented by the key on which they are found. For example: the shaded graphic squares found on keys "H" and "A" are uploaded as "H" and "A". The context again usually supplies enough indication that a block graphic character is required and there is the added bonus of not needing to hunt around for the key it is on or having to figure how many graphic characters you have in a row. Also again any problem lines can be annotated once the listing is loaded into M-Script.

3) The rogue TS 1000 BASIC tokens "" (shifted-Q), SLOW, FAST, SCROLL, UNPLOT are now represented by: !, [,], %, &. The beauty of this is that once the listing is in M-Script there is no need to chase around looking for all the TS 1000 tokens that do not upload as TS 2068 tokens. All you have to do is enter into edit mode in M-Script and use the global search and replace routine to change all the !'s into ""'s the ['s into SLOW's and so on. As long as you are careful to place the cursor at the beginning of the file before engaging the search and replace routine you do not need to worry about missing any occurance of the symbol your are replacing because M-Script will replace them automatically. While you are at it you can change ^ into ** and GO TO, GO SUB and RANDOMIZE into GOTO, GOSUB and RAND to complete the listing.

Here is the listing of the First Loader program incorporating the changes I have made to it. I would like to finish by welcoming you all to the world of neat and organized program listing.

First Loader
Program Listing
(Modification by D. Solly)

```
   1 REM This is a modification of an
original program by David Ornstein by David
Solly to be used in conjunction with Basic2
by Michael E. Carver to produce ASCII text
versions of TS 1000 or ZX81 BASIC program
listings which can be edited through the use
of Jack Dohany's version of M-Script.
   2 REM The original version of this program
was written by David Ornstein for the
conversion of TS 1000 or ZX81 programs into
TS 2068 programs. Program first appeared in
Syntax (ISSN: 0273-2696): v. 4, no. 12 1983
pp. 14-18 and v. 5, no. 1 1984 p. 2.
   3 REM To create the machine code type RUN.
The resulting machine code program is saved
by a GO TO 9050. Instructions for the use of
the machine code program given below.
   5 REM Instructions: Enter CLEAR 43999.
Load the First Loader machine code program.
When the program has loaded type RANDOMIZE
USER 45000 but do not ENTER. Put the ZX81 (TS
1000) program into the recorder and locate
the silent passage just before the program
begins. Using this as a starting point
restart the tape and hit ENTER. When the
program has stoped with OK the resulting TS
2068 program may be LISTed and SAVEd in the
usual way. The uploaded program produced from
this version of First Loader is not designed
to be RUN but as an initial step towards
conversion into ASCII Text to be manipulated
with M-Script.
   6 REM
  50 CLEAR 43999: RESTORE 100
  60 FOR a=44000 TO 44255
  70 READ n: POKE a,n
  80 NEXT a
 100 DATA 32,49,50,55,52
 105 DATA 53,84,69,65,68
 110 DATA 83,34,96,36,58
 115 DATA 63,40,41,62,60
 120 DATA 61,43,45,42,47
 125 DATA 59,44,46,48,49
 130 DATA 50,51,52,53,54
 135 DATA 55,56,57,65,66
 140 DATA 67,68,69,70,71
 145 DATA 72,73,74,75,76
 150 DATA 77,78,79,80,81
```

```
155 DATA 82,83,84,85,86
160 DATA 87,88,89,90,165
165 DATA 166,167,32,32,32
170 DATA 32,32,32,32,32
175 DATA 32,32,32,32,32
180 DATA 32,32,32,32,32
185 DATA 32,32,32,32,32
190 DATA 32,32,32,32,32
195 DATA 32,32,32,32,32
200 DATA 32,32,32,32,32
205 DATA 32,32,32,32,32
210 DATA 32,32,32,32,32
215 DATA 32,32,32,13,32
220 DATA 32,32,32,32,32
225 DATA 32,14,32,64,81
230 DATA 87,54,82,56,89
235 DATA 51,72,71,70,34
240 DATA 96,36,58,63,40
245 DATA 41,62,60,61,43
250 DATA 45,42,47,59,44
255 DATA 46,48,49,50,51
260 DATA 52,53,54,55,56
265 DATA 57,97,98,99,100
270 DATA 101,102,103,104,105
275 DATA 106,107,108,109,110
280 DATA 111,112,113,114,115
285 DATA 116,117,118,119,120
290 DATA 121,122,33,172,173
295 DATA 32,175,176,177,178
300 DATA 179,180,181,182,183
305 DATA 184,185,186,187,188
310 DATA 189,190,192,193,194
315 DATA 195,94,197,198,199
320 DATA 200,201,203,204,205
325 DATA 224,225,226,91,93
330 DATA 230,37,232,233,234
335 DATA 235,236,237,238,239
340 DATA 240,241,242,243,244
345 DATA 245,246,247,248,249
350 DATA 250,251,38,253,254
355 DATA 255
360 REM End of first section of code
400 RESTORE 500
405 FOR a=45000 TO 45233
420 READ n: POKE a,n
430 NEXT a
500 DATA 205,60,176,205,222
505 DATA 175,6,7,54,118
510 DATA 35,16,251,205,89
515 DATA 176,205,142,176,201
520 DATA 207,3,33,9,192
```

```
 525 DATA 17,12,0,25,54
 530 DATA 254,167,237,82,205
 535 DATA 251,175,113,124,254
 540 DATA 0,40,233,205,75
 545 DATA 176,205,46,176,24
 550 DATA 239,14,1,6,0
 555 DATA 205,68,176,62,127
 560 DATA 219,254,31,48,49
 565 DATA 23,23,23,56,2
 570 DATA 24,239,213,30,148
 575 DATA 6,26,29,219,254
 580 DATA 23,23,203,123,123
 585 DATA 56,244,16,244,209
 590 DATA 32,4,254,86,48
 595 DATA 213,63,203,17,48
 600 DATA 208,201,35,235,42
 605 DATA 20,192,55,237,82
 610 DATA 235,208,193,201,207
 615 DATA 12,205,251,175,203
 620 DATA 121,40,249,201,120
 625 DATA 238,2,211,254,71
 630 DATA 201,17,21,192,122
 635 DATA 188,192,123,189,192
 640 DATA 126,198,128,119,201
 645 DATA 33,125,192,126,254
 650 DATA 118,200,205,101,176
 655 DATA 24,247,17,4,0
 660 DATA 25,62,127,219,254
 665 DATA 31,48,202,126,254
 670 DATA 126,32,8,54,14
 675 DATA 17,6,0,25,24
 680 DATA 236,229,17,224,171
 685 DATA 38,0,111,25,126
 690 DATA 225,119,35,254,13
 695 DATA 32,220,201,17,125
 700 DATA 192,167,237,82,68
 705 DATA 77,197,42,75,92
 710 DATA 43,237,91,83,92
 715 DATA 213,205,187,18,209
 720 DATA 237,83,83,92,193
 725 DATA 35,35,235,33,125
 730 DATA 192,237,176,201
 800 STOP
9000 REM CASSETTE SAVE
9010 SAVE "Lpfl.B1"
9020 CLS : PRINT "Verification: Rewind and
play."
9030 VERIFY ""
9040 STOP
9050 REM MC SAVE
9070 SAVE "Lpfl.C1"CODE 44000,1250
```

```
9085 CLS : PRINT "Verification: Rewind and
play."
9090 VERIFY ""CODE
```